

WEST Search History

DATE: Monday, May 02, 2005

Hide? Set Name Query**Hit Count***DB=USPT,EPAB,JPAB,DWPI,TDBD; PLUR=YES; OP=OR*

<input type="checkbox"/>	L15	(compress\$4 near3 image near3 kernel)	9
<input type="checkbox"/>	L14	(compress\$4 near3 image near3 kernel) same reboot\$4	0
<input type="checkbox"/>	L13	L9 same ((based or depend\$4 or accord\$4) near5 image)	2
<input type="checkbox"/>	L12	L9 same ((based or depend\$4 or accord\$4) near5 copy)	0
<input type="checkbox"/>	L11	L9 same ((based or depend\$4 or accord\$4) near5 kernel)	4
<input type="checkbox"/>	L10	L9 same (based or depend\$4 or accord\$4)	633
<input type="checkbox"/>	L9	((reload\$4 or replac\$4) near5 (os or (operating adj system)))	14049
<input type="checkbox"/>	L8	l5 same ((replac\$4 or reload\$4) near5 (os or (operating adj system)))	1
<input type="checkbox"/>	L7	l5 same (reload\$4 near5 (os or (operating adj system)))	1
<input type="checkbox"/>	L6	L5 and l1	0
<input type="checkbox"/>	L5	(warm near2 reboot\$4)	57
<input type="checkbox"/>	L4	L1 same copy	0
<input type="checkbox"/>	L3	L1 same image	0
<input type="checkbox"/>	L2	L1 same kernel	0
<input type="checkbox"/>	L1	(replac\$4 near3 (whole or entire) near3 (os or (operating adj system)))	16

END OF SEARCH HISTORY

[First Hit](#) [Fwd Refs](#)[Previous Doc](#)[Next Doc](#)[Go to Doc#](#)

End of Result Set



Generate Collection

Print

L7: Entry 1 of 1

File: USPT

Aug 10, 1999

DOCUMENT-IDENTIFIER: US 5935242 A

TITLE: Method and apparatus for initializing a device

Brief Summary Text (11):

Embodiments of the present invention provide a system for controlling the rebooting of a device such as a computer. When a warm reboot is triggered, the invention provides a mechanism for determining whether a valid copy of the operating system and related files are already stored in a memory device in the computer. The system avoids reloading the entire operating system if a valid copy of the operating system is already stored in the computer. The system does not require the additional cost and complexity associated with systems using Flash ROMs. This is particularly important in cost-sensitive devices such as network-booted computers and other network-booted devices.

[Previous Doc](#)[Next Doc](#)[Go to Doc#](#)

[First Hit](#) [Fwd Refs](#)[Previous Doc](#)[Next Doc](#)[Go to Doc#](#)

Generate Collection

Print

L1: Entry 6 of 16

File: USPT

Sep 9, 1997

DOCUMENT-IDENTIFIER: US 5666293 A

TITLE: Downloading operating system software through a broadcast channel

Detailed Description Text (70):

In the preferred embodiment, the operating system for the DET 102 includes a version of a PC type operating system, e.g. OS-9. In addition, the operating system for the DET 102 includes the various drivers necessary for the DET microprocessor 110 to operate the associated peripherals, e.g. the Digital Audio/Video Processor 125, the Personal Computer Memory Card Industry Association (PCMCIA) port 155, the RS-232 transceiver 151, etc. The set-top operating system also includes the resident cable television emulation software, i.e. as needed to facilitate reception of broadcast programs through the particular network. This operating system is stored in a portion of the non-volatile RAM 121 having a relatively low level of protection. When a new operating system is installed, as discussed more fully below, the new operating system replaces the entire operating system previously stored in the non-volatile RAM. The level of protection here provided enables rewriting the operating system using a broadcast channel download procedure, however, there is sufficient protection to limit storage to only acceptable software from an authorized provider.

[Previous Doc](#)[Next Doc](#)[Go to Doc#](#)

[First Hit](#) [Fwd Refs](#)[Previous Doc](#)[Next Doc](#)[Go to Doc#](#)

Generate Collection

Print

L11: Entry 1 of 4

File: USPT

Dec 19, 2000

DOCUMENT-IDENTIFIER: US 6163812 A

TITLE: Adaptive fast path architecture for commercial operating systems and information server applications

Brief Summary Text (4):

Extensibility is found in most modern operating systems, but it is often limited to supporting new devices, not specializing the usage of system resources. One technique exposes operating system policies to applications via event/handler extensions. This is described in B. N. Bershad et al., Extensibility, Safety and Performance in the Operating System, Proceedings of the Fifteenth Symposium on Operating System Principles, 1995. A second technique provides adaptive in-kernel policies that manage resources based on application usage patterns. This is described in C. Pu et al., Optimistic Incremental Specialization: Streamlining a Commercial Operating System, Proceedings of the Fifteenth Symposium on Operating System Principles, 1995; and P. Cao et al., Implementation and Performance of Application-Controlled File Caching, Proceedings of the First Symposium on Operating Systems Design and Implementation (OSDI '94), pp. 165-177, November 1994. A third technique replaces underlying components of the operating system with new components referred to as library operating systems. This is described in D. R. Engler et al., Exokernel: An Operating System Architecture for Application-Level Resource Management, Proceedings of the Fifteenth Symposium on Operating System Principles, 1995.

[Previous Doc](#)[Next Doc](#)[Go to Doc#](#)

Go to Doc#

Print

Sep 21, 1999

Go to Doc#

Go to Doc#

☐ Generate Collection ☐ Print

Aug 23, 1994

TITLE: Method and apparatus for processing multiple file system server requests in a data processing network

In the prior art a user desiring to utilize a particular server device must ensure that the appropriate File System Driver has been loaded. Subsequently, if a user desires to utilize an alternate server it is necessary to reload the operating system and select an alternate File System Driver. Recently, the International Business Machines Corporation OS/2 Operating System Version 1.3 has demonstrated the utilization of dual File System Driver support. This operating system supports both the IBM Network File System Driver and the Novell Network File System Driver by changing all path-based API's kernel routines to permit failing calls for a server to one File System Driver to be automatically be sent to the alternate File System Driver. While this represents an increase in network efficiency, this approach only applies to the IBM/Novell File System Drivers and does not accommodate concurrent utilization of both File System Drivers.

Go to Doc#